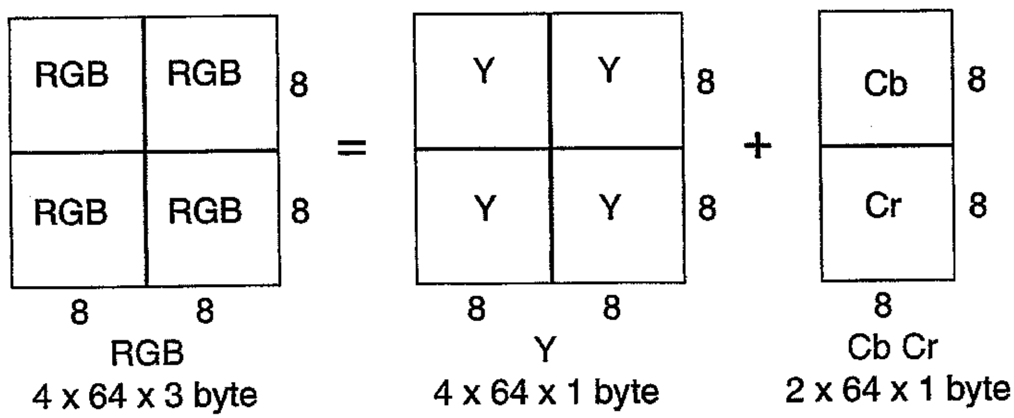
Second Declaration of George T. Ligler

Exhibit 16

JPEG Compression

The image is divided in 16x16 pixel boxes, each is subdivided in 8x8 boxes.



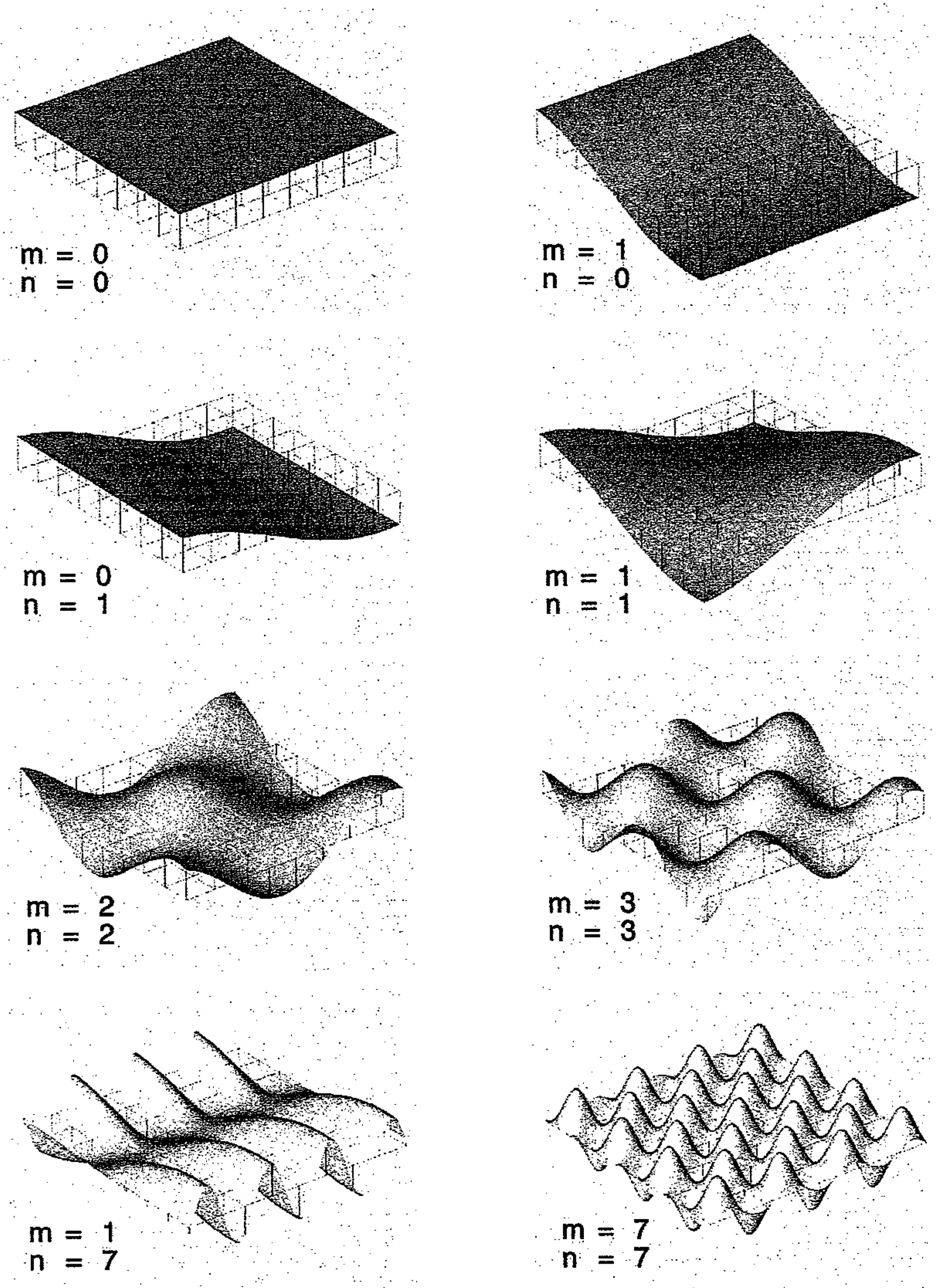
The Luminance Y is calculated for each 8x8 box, but the chrominance Cb,Cr is taken from each second row and each second column only, in fact by averaging four values for one result. This is already a compression ratio of two.

$$
\begin{aligned}
Y &= 0.299\,R + 0.587\,G + 0.114\,B \\
Cb &= -0.169\,R - 0.331\,G + 0.500\,B \\
Cr &= 0.500\,R - 0.419\,G - 0.081\,B
\end{aligned}
$$

Y is a substitute for the Luminance (based on inverse gamma-encoded data), but the residual colors Cb and Cr do not have any physical meaning.
Then the content of each 8x8 box, which we call here C_{xy}, altogether 64 values C_{00} to C_{77}, is treated by the same mathematical algorithm, the Discrete Cosine Transformation DCT.
On the next page we see a subset of 64 functions F_{mn}. F_{00} is a constant function, F_{10} has a half-cosine slope in x-direction and F_{01} in y-direction. F_{77} is the most complex function, 4 cosine periods in each direction.
Then it has to be calculated how much each function contributes to the discrete landscape C_{xy}. Result is a set of 64 coefficients K_{mn}.
K_{00} is the DC- or mean value. K_{77} the highest frequency content.

The DCT is lossless and reversible, besides some minor round-off errors because the coefficients K_{mn} are stored ShortInt -128 to +127.
In any continuous tone image the high frequencies are rare, with a few exceptions here and there.

Discrete Cosine Transformation / Zoom 200%



AX205630

This is a table of 64 DCT coefficients, in ShortInt -128...+127 .

The low frequency part is top left, the high frequency part bottom right.
A linear fading or shaded color bar in vertical direction has mainly coefficients K_{00} and K_{01}.
Minor corrections are necessary because the linear slope needs to be replaced by cosines.

K_{00}	K_{10}	K_{20}	K_{30}	K_{40}	K_{50}	K_{60}	K_{70}
K_{01}	K_{11}	K_{21}	K_{31}	K_{41}	K_{51}	K_{61}	K_{71}
K_{02}	K_{12}	K_{22}	K_{32}	K_{42}	K_{52}	K_{62}	K_{72}
K_{03}	K_{13}	K_{23}	K_{33}	K_{43}	K_{53}	K_{63}	K_{73}
K_{04}	K_{14}	K_{24}	K_{34}	K_{44}	K_{54}	K_{64}	K_{74}
K_{05}	K_{15}	K_{25}	K_{35}	K_{45}	K_{55}	K_{65}	K_{75}
K_{06}	K_{16}	K_{26}	K_{36}	K_{46}	K_{56}	K_{66}	K_{76}
K_{07}	K_{17}	K_{27}	K_{37}	K_{47}	K_{57}	K_{67}	K_{77}

DCT - Coefficients

Then each coefficient is divided by values from the so called Quantization Tables.
These are different for Y and Cb, Cr.
Because the results are always rounded off to ShortInt, we get many small numbers and plenty zeros in the results, which replace the original DCT coefficients.
A slightly disturbed symmetry can prevent from generating patterns.

16	11	10					61
12	12	14					55
14	13	16					56
72	92	95					99

Y - Quantization Table

Especially the high frequency part bottom right consists mainly of zeros.
The Quantization Values are larger for Cb,Cr than for Y, thus the color part is treated once again very badly.
The tables were probably found by experiments by the JPEG experts.

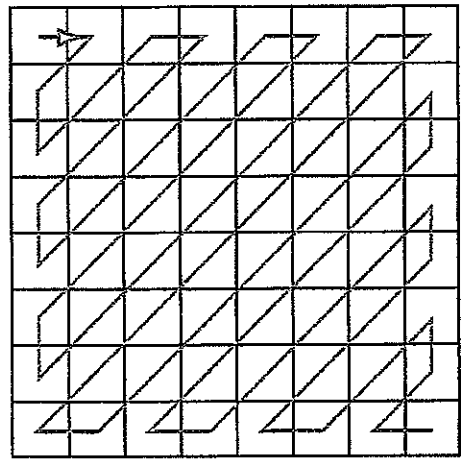
Finally, each 8x8 box contains only a few significant small numbers. This is the lossy compression.

17	18	24	47	99			99
18	21	26	66	99			99
24	26	56	99				99
47	66	99					
99	99						
99	99	99					99

CbCr - Quantization Table

AX205631

The divided DCT coefficients are scanned in ZigZag order.
First the low frequency part, finally the high frequency part.
It can be expected, that here are many sequential zeros.



ZigZag Table

The sequential zeros are lossless compressed by Runlength Encoding RLE. For example 0,0,0,0,0 will deliver 0,5 , thus a much shorter Code.

Then we have to take into account, that additionally many small numbers appear, e.g. -2, -1, +1, +2 . The greatest number is approximately 128/16= 8.

Now the lossless Huffman Encoding is applied. Small numbers are coded by few bits, large number by more bits, instead of all numbers by one Byte. This is controlled by a Huffman Table. Huffman and Runlength together pack the divided DCT coefficients highly compressed but lossless.
Of course it is a rather complex task to apply Huffman because a sequential bitstream needs to packed by number codes of different bitlengths.

The receiver finds in the File Header two Quantization Tables and two Huffman Tables. The Color Matrix is a standard, the ZigZag Table as well.

The Decoding works exactly in opposite direction:
Decode Huffman Bitstream. Unpack Anti-ZigZag. Multiply by Quantization. Apply Inverse DCT.
Repeat this until six 8x8 blocks for one 16x16 RGB block are found.
Apply color doubling for rows and columns. Apply Inverse Color Matrix. Assemble 16x16 RGB block. Draw block.
The compression ratio is controlled by the Quantization Tables. Multiplying the tables by 0.5 will improve the quality but increase the filesize. Multiplying by 2 will do just the opposite. Appropriate factors are found by experiments.

The Compression Ratio is the quotient
C = UncompressedFileSize / CompressedFileSize

C = 10
Excellent quality; compression cannot be detected by sharpeners
C = 20...30
Good quality; compression can be detected by sharpeners
C = 50
Bad quality; boxes visible

Each JPEG File needs at least 0.5 kByte for the header and the tables in the author´s program. In PhotoShop the minimum size is eventually 4kByte, if additional information is embedded (preview, color management).
For Web applications it is recommended to avoid additional information by 'Save for Web' in Photoshop.

JPEG compression tends to show softening effects. Nevertheless single sharp lines are not ignored. Sharpening in advance to the compression can improve the quality, but the file size will be increased because high frequency parts are now essential.

Though JPEG is based on TrueColor, standard Encoders and Decoders can be used for grayscale images. In a color image,the gray part may anyway consume 75% of the file size, because of lower compression.

This report by Gernot Hoffmann is mainly based on the Diploma Thesis of Hermann Hildebrandt (1999) and the cooperation with him.
Later, the author had replaced all kernels by Assembler procedures and functions, even for the floating point operations, mainly the DCT.
All tests for encoding and decoding were performed using PhotoShop and some other Windows programs.
The JPEG package is a part of the non-commercial image processing program ZEBRA. The illustration was generated by the experimental Computer Graphics System ZEFIR.

Gernot Hoffmann
September 18 / 2003

Website

Second Declaration of George T. Ligler

Exhibit 17

Jpeg 2000 Compression
Access Interactive Design Tools, Evaluation Tools & Software.

Jpeg Source Code
High performance code for audio, video, & graphics. Code Samples.

JPEG 2000 Compression SDK
Components, Libs, Multiple OS's Add JPEG 2000 to Software Apps

Convert File Format
Free Software. Instant [
Pdf, html, txt, tif, jpg, g

Ads by

[Previous] [Next]

JPEG Compression

One of the hottest topics in image compression technology today is JPEG. The acronym JPEG stands for the Joint Photographic Experts Group, a standards committee that had its origins within the International Standard Organization (ISO). In 1982, the ISO formed the Photographic Experts Group (PEG) to research methods of transmitting video, still images, and text over ISDN (Integrated Services Digital Network) lines. PEG's goal was to produce a set of industry standards for the transmission of graphics and image data over digital communications networks.

In 1986, a subgroup of the CCITT began to research methods of compressing color and gray-scale data for facsimile transmission. The compression methods needed for color facsimile systems were very similar to those being researched by PEG. It was therefore agreed that the two groups should combine their resources and work together toward a single standard.

In 1987, the ISO and CCITT combined their two groups into a joint committee that would research and produce a single standard of image data compression for both organizations to use. This new committee was JPEG.

Although the creators of JPEG might have envisioned a multitude of commercial applications for JPEG technology, a consumer public made hungry by the marketing promises of imaging and multimedia technology are benefiting greatly as well. Most previously developed compression methods do a relatively poor job of compressing continuous-tone image data; that is, images containing hundreds or thousands of colors taken from real-world subjects. And very few file formats can support 24-bit raster images.

GIF, for example, can store only images with a maximum pixel depth of eight bits, for a maximum of 256 colors. And its LZW compression algorithm does not

work very well on typical scanned image data. The low-level noise commonly found in such data defeats LZW's ability to recognize repeated patterns.

Both TIFF and BMP are capable of storing 24-bit data, but in their pre-JPEG versions are capable of using only encoding schemes (LZW and RLE, respectively) that do not compress this type of image data very well.

JPEG provides a compression method that is capable of compressing continuous-tone image data with a pixel depth of 6 to 24 bits with reasonable speed and efficiency. And although JPEG itself does not define a standard image file format, several have been invented or modified to fill the needs of JPEG data storage.

JPEG in Perspective

Unlike all of the other compression methods described so far in this chapter, JPEG is not a single algorithm. Instead, it may be thought of as a toolkit of image compression methods that may be altered to fit the needs of the user. JPEG may be adjusted to produce very small, compressed images that are of relatively poor quality in appearance but still suitable for many applications. Conversely, JPEG is capable of producing very high-quality compressed images that are still far smaller than the original uncompressed data.

JPEG is also different in that it is primarily a lossy method of compression. Most popular image format compression schemes, such as RLE, LZW, or the CCITT standards, are lossless compression methods. That is, they do not discard any data during the encoding process. An image compressed using a lossless method is guaranteed to be identical to the original image when uncompressed.

Lossy schemes, on the other hand, throw useless data away during encoding. This is, in fact, how lossy schemes manage to obtain superior compression ratios over most lossless schemes. JPEG was designed specifically to discard information that the human eye cannot easily see. Slight changes in color are not perceived well by the human eye, while slight changes in intensity (light and dark) are. Therefore JPEG's lossy encoding tends to be more frugal with the gray-scale part of an image and to be more frivolous with the color.

JPEG was designed to compress color or gray-scale continuous-tone images of real-world subjects: photographs, video stills, or any complex graphics that resemble natural subjects. Animations, ray tracing, line art, black-and-white documents, and typical vector graphics don't compress very well under JPEG and shouldn't be expected to. And, although JPEG is now used to provide motion video compression, the standard makes no special provision for such an application.

The fact that JPEG is lossy and works only on a select type of image data might make you ask, "Why bother to use it?" It depends upon your needs. JPEG is an excellent way to store 24-bit photographic images, such as those used in

imaging and multimedia applications. JPEG 24-bit (16 million color) images are superior in appearance to 8-bit (256 color) images on a VGA display and are at their most spectacular when using 24-bit display hardware (which is now quite inexpensive).

The amount of compression achieved depends upon the content of the image data. A typical photographic-quality image may be compressed from 20:1 to 25:1 without experiencing any noticeable degradation in quality. Higher compression ratios will result in image files that differ noticeably from the original image but still have an overall good image quality. And achieving a 20:1 or better compression ratio in many cases not only saves disk space, but also reduces transmission time across data networks and phone lines.

An end user can "tune" the quality of a JPEG encoder using a parameter sometimes called a *quality setting* or a *Q factor*. Although different implementations have varying scales of Q factors, a range of 1 to 100 is typical. A factor of 1 produces the smallest, worst quality images; a factor of 100 produces the largest, best quality images. The optimal Q factor depends on the image content and is therefore different for every image. The art of JPEG compression is finding the lowest Q factor that produces an image that is visibly acceptable, and preferably as close to the original as possible.

The JPEG library supplied by the Independent JPEG Group uses a quality setting scale of 1 to 100. To find the optimal compression for an image using the JPEG library, follow these steps:

1. Encode the image using a quality setting of 75 (-Q 75).
2. If you observe unacceptable defects in the image, increase the value, and re-encode the image.
3. If the image quality is acceptable, decrease the setting until the image quality is barely acceptable. This will be the optimal quality setting for this image.
4. Repeat this process for every image you have (or just encode them all using a quality setting of 75).

JPEG isn't always an ideal compression solution. There are several reasons:

- As we have said, JPEG doesn't fit every compression need. Images containing large areas of a single color do not compress very well. In fact, JPEG will introduce "artifacts" into such images that are visible against a flat background, making them considerably worse in appearance than if you used a conventional lossless compression method. Images of a "busier" composition contain even worse artifacts, but they are considerably less noticeable against the image's more complex background.

- JPEG can be rather slow when it is implemented only in software. If fast decompression is required, a hardware-based JPEG solution is your best bet, unless you are willing to wait for a faster software-only solution to come along or buy a faster computer.

- JPEG is not trivial to implement. It is not likely you will be able to sit down and write your own JPEG encoder/decoder in a few evenings. We recommend that you obtain a third-party JPEG library, rather than writing your own.

- JPEG is not supported by very many file formats. The formats that do support JPEG are all fairly new and can be expected to be revised at frequent intervals.

Baseline JPEG

The JPEG specification defines a minimal subset of the standard called baseline JPEG, which all JPEG-aware applications are required to support. This baseline uses an encoding scheme based on the Discrete Cosine Transform (DCT) to achieve compression. DCT is a generic name for a class of operations identified and published some years ago. DCT-based algorithms have since made their way into various compression methods.

DCT-based encoding algorithms are always lossy by nature. DCT algorithms are capable of achieving a high degree of compression with only minimal loss of data. This scheme is effective only for compressing continuous-tone images in which the differences between adjacent pixels are usually small. In practice, JPEG works well only on images with depths of at least four or five bits per color channel. The baseline standard actually specifies eight bits per input sample. Data of lesser bit depth can be handled by scaling it up to eight bits per sample, but the results will be bad for low-bit-depth source data, because of the large jumps between adjacent pixel values. For similar reasons, colormapped source data does not work very well, especially if the image has been dithered.

The JPEG compression scheme is divided into the following stages:

1. Transform the image into an optimal color space.

2. Downsample chrominance components by averaging groups of pixels together.

3. Apply a Discrete Cosine Transform (DCT) to blocks of pixels, thus removing redundant image data.

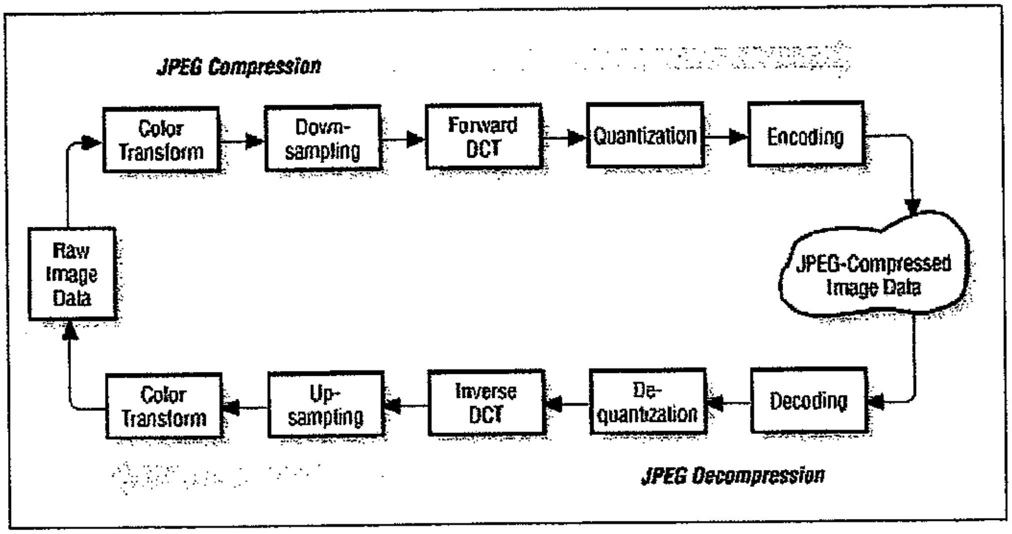
4. Quantize each block of DCT coefficients using weighting functions optimized for the human eye.

AX205617

5. Encode the resulting coefficients (image data) using a Huffman variable word-length algorithm to remove redundancies in the coefficients.

Figure 9-11 summarizes these steps, and the following subsections look at each of them in turn. Note that JPEG decoding performs the reverse of these steps.

Figure 9-11: JPEG compression and decompression



Transform the image

The JPEG algorithm is capable of encoding images that use any type of color space. JPEG itself encodes each component in a color model separately, and it is completely independent of any color-space model, such as RGB, HSI, or CMY. The best compression ratios result if a luminance/chrominance color space, such as YUV or YCbCr, is used. (See Chapter 2 for a description of these color spaces.)

Most of the visual information to which human eyes are most sensitive is found in the high-frequency, gray-scale, luminance component (Y) of the YCbCr color space. The other two chrominance components (Cb and Cr) contain high-frequency color information to which the human eye is less sensitive. Most of this information can therefore be discarded.

In comparison, the RGB, HSI, and CMY color models spread their useful visual image information evenly across each of their three color components, making the selective discarding of information very difficult. All three color components would need to be encoded at the highest quality, resulting in a poorer compression ratio. Gray-scale images do not have a color space as such and therefore do not require transforming.

Downsample chrominance components

The simplest way of exploiting the eye's lesser sensitivity to chrominance information is simply to use fewer pixels for the chrominance channels. For

example, in an image nominally 1000x1000 pixels, we might use a full 1000x1000 luminance pixels but only 500x500 pixels for each chrominance component. In this representation, each chrominance pixel covers the same area as a 2x2 block of luminance pixels. We store a total of six pixel values for each 2x2 block (four luminance values, one each for the two chrominance channels), rather than the twelve values needed if each component is represented at full resolution. Remarkably, this 50 percent reduction in data volume has almost no effect on the perceived quality of most images. Equivalent savings are not possible with conventional color models such as RGB, because in RGB each color channel carries some luminance information and so any loss of resolution is quite visible.

When the uncompressed data is supplied in a conventional format (equal resolution for all channels), a JPEG compressor must reduce the resolution of the chrominance channels by *downsampling*, or averaging together groups of pixels. The JPEG standard allows several different choices for the sampling ratios, or relative sizes, of the downsampled channels. The luminance channel is always left at full resolution (1:1 sampling). Typically both chrominance channels are downsampled 2:1 horizontally and either 1:1 or 2:1 vertically, meaning that a chrominance pixel covers the same area as either a 2x1 or a 2x2 block of luminance pixels. JPEG refers to these downsampling processes as 2h1v and 2h2v sampling, respectively.

Another notation commonly used is 4:2:2 sampling for 2h1v and 4:2:0 sampling for 2h2v; this notation derives from television customs (color transformation and downsampling have been in use since the beginning of color TV transmission). 2h1v sampling is fairly common because it corresponds to National Television Standards Committee (NTSC) standard TV practice, but it offers less compression than 2h2v sampling, with hardly any gain in perceived quality.

Apply a Discrete Cosine Transform

The image data is divided up into 8x8 blocks of pixels. (From this point on, each color component is processed independently, so a "pixel" means a single value, even in a color image.) A DCT is applied to each 8x8 block. DCT converts the spatial image representation into a frequency map: the low-order or "DC" term represents the average value in the block, while successive higher-order ("AC") terms represent the strength of more and more rapid changes across the width or height of the block. The highest AC term represents the strength of a cosine wave alternating from maximum to minimum at adjacent pixels.

The DCT calculation is fairly complex; in fact, this is the most costly step in JPEG compression. The point of doing it is that we have now separated out the high- and low-frequency information present in the image. We can discard high-frequency data easily without losing low-frequency information. The DCT step itself is lossless except for roundoff errors.

Quantize each block

To discard an appropriate amount of information, the compressor divides each DCT output value by a "quantization coefficient" and rounds the result to an integer. The larger the quantization coefficient, the more data is lost, because the actual DCT value is represented less and less accurately. Each of the 64 positions of the DCT output block has its own quantization coefficient, with the higher-order terms being quantized more heavily than the low-order terms (that is, the higher-order terms have larger quantization coefficients). Furthermore, separate quantization tables are employed for luminance and chrominance data, with the chrominance data being quantized more heavily than the luminance data. This allows JPEG to exploit further the eye's differing sensitivity to luminance and chrominance.

It is this step that is controlled by the "quality" setting of most JPEG compressors. The compressor starts from a built-in table that is appropriate for a medium-quality setting and increases or decreases the value of each table entry in inverse proportion to the requested quality. The complete quantization tables actually used are recorded in the compressed file so that the decompressor will know how to (approximately) reconstruct the DCT coefficients.

Selection of an appropriate quantization table is something of a black art. Most existing compressors start from a sample table developed by the ISO JPEG committee. It is likely that future research will yield better tables that provide more compression for the same perceived image quality. Implementation of improved tables should not cause any compatibility problems, because decompressors merely read the tables from the compressed file; they don't care how the table was picked.

Encode the resulting coefficients

The resulting coefficients contain a significant amount of redundant data. Huffman compression will losslessly remove the redundancies, resulting in smaller JPEG data. An optional extension to the JPEG specification allows arithmetic encoding to be used instead of Huffman for an even greater compression ratio. (See the section called "JPEG Extensions (Part 1)" below.) At this point, the JPEG data stream is ready to be transmitted across a communications channel or encapsulated inside an image file format.

JPEG Extensions (Part 1)

What we have examined thus far is only the baseline specification for JPEG. A number of extensions have been defined in Part 1 of the JPEG specification that provide progressive image buildup, improved compression ratios using arithmetic encoding, and a lossless compression scheme. These features are beyond the needs of most JPEG implementations and have therefore been defined as "not required to be supported" extensions to the JPEG standard.

Progressive image buildup

AX205620

Progressive image buildup is an extension for use in applications that need to receive JPEG data streams and display them on the fly. A baseline JPEG image can be displayed only after all of the image data has been received and decoded. But some applications require that the image be displayed after only some of the data is received. Using a conventional compression method, this means displaying the first few scan lines of the image as it is decoded. In this case, even if the scan lines were interlaced, you would need at least 50 percent of the image data to get a good clue as to the content of the image. The progressive buildup extension of JPEG offers a better solution.

Progressive buildup allows an image to be sent in layers rather than scan lines. But instead of transmitting each bitplane or color channel in sequence (which wouldn't be very useful), a succession of images built up from approximations of the original image are sent. The first scan provides a low-accuracy representation of the entire image--in effect, a very low-quality JPEG compressed image. Subsequent scans gradually refine the image by increasing the effective quality factor. If the data is displayed on the fly, you would first see a crude, but recognizable, rendering of the whole image. This would appear very quickly because only a small amount of data would need to be transmitted to produce it. Each subsequent scan would improve the displayed image's quality one block at a time.

A limitation of progressive JPEG is that each scan takes essentially a full JPEG decompression cycle to display. Therefore, with typical data transmission rates, a very fast JPEG decoder (probably specialized hardware) would be needed to make effective use of progressive transmission.

A related JPEG extension provides for hierarchical storage of the same image at multiple resolutions. For example, an image might be stored at 250x250, 500x500, 1000x1000, and 2000x2000 pixels, so that the same image file could support display on low-resolution screens, medium-resolution laser printers, and high-resolution imagesetters. The higher-resolution images are stored as differences from the lower-resolution ones, so they need less space than they would need if they were stored independently. This is not the same as a progressive series, because each image is available in its own right at the full desired quality.

Arithmetic encoding

The baseline JPEG standard defines Huffman compression as the final step in the encoding process. A JPEG extension replaces the Huffman engine with a binary arithmetic entropy encoder. The use of an arithmetic coder reduces the resulting size of the JPEG data by a further 10 percent to 15 percent over the results that would be achieved by the Huffman coder. With no change in resulting image quality, this gain could be of importance in implementations where enormous quantities of JPEG images are archived.

AX205621

Arithmetic encoding has several drawbacks:

- Not all JPEG decoders support arithmetic decoding. Baseline JPEG decoders are required to support only the Huffman algorithm.
- The arithmetic algorithm is slower in both encoding and decoding than Huffman.
- The arithmetic coder used by JPEG (called a *Q-coder*) is owned by IBM and AT&T. (Mitsubishi also holds patents on arithmetic coding.) You must obtain a license from the appropriate vendors if their Q-coders are to be used as the back end of your JPEG implementation.

Lossless JPEG compression

A question that commonly arises is "At what Q factor does JPEG become lossless?" The answer is "never." Baseline JPEG is a lossy method of compression regardless of adjustments you may make in the parameters. In fact, DCT-based encoders are always lossy, because roundoff errors are inevitable in the color conversion and DCT steps. You can suppress deliberate information loss in the downsampling and quantization steps, but you still won't get an exact recreation of the original bits. Further, this minimum-loss setting is a very inefficient way to use lossy JPEG.

The JPEG standard does offer a separate lossless mode. This mode has nothing in common with the regular DCT-based algorithms, and it is currently implemented only in a few commercial applications. JPEG lossless is a form of Predictive Lossless Coding using a 2D Differential Pulse Code Modulation (DPCM) scheme. The basic premise is that the value of a pixel is combined with the values of up to three neighboring pixels to form a predictor value. The predictor value is then subtracted from the original pixel value. When the entire bitmap has been processed, the resulting predictors are compressed using either the Huffman or the binary arithmetic entropy encoding methods described in the JPEG standard.

Lossless JPEG works on images with 2 to 16 bits per pixel, but performs best on images with 6 or more bits per pixel. For such images, the typical compression ratio achieved is 2:1. For image data with fewer bits per pixels, other compression schemes do perform better.

JPEG Extensions (Part 3)

The following JPEG extensions are described in Part 3 of the JPEG specification.

Variable quantization

Variable quantization is an enhancement available to the quantization procedure of DCT-based processes. This enhancement may be used with any of the DCT-

based processes defined by JPEG with the exception of the baseline process.

The process of quantization used in JPEG quantizes each of the 64 DCT coefficients using a corresponding value from a quantization table. Quantization values may be redefined prior to the start of a scan but must not be changed once they are within a scan of the compressed data stream.

Variable quantization allows the scaling of quantization values within the compressed data stream. At the start of each 8x8 block is a quantizer scale factor used to scale the quantization table values within an image component and to match these values with the AC coefficients stored in the compressed data. Quantization values may then be located and changed as needed.

Variable quantization allows the characteristics of an image to be changed to control the quality of the output based on a given model. The variable quantizer can constantly adjust during decoding to provide optimal output.

The amount of output data can also be decreased or increased by raising or lowering the quantizer scale factor. The maximum size of the resulting JPEG file or data stream may be imposed by constant adaptive adjustments made by the variable quantizer.

The variable quantization extension also allows JPEG to store image data originally encoded using a variable quantization scheme, such as MPEG. For MPEG data to be accurately transcoded into another format, the other format must support variable quantization to maintain a high compression ratio. This extension allows JPEG to support a data stream originally derived from a variably quantized source, such as an MPEG I-frame.

Selective refinement

Selective refinement is used to select a region of an image for further enhancement. This enhancement improves the resolution and detail of a region of an image. JPEG supports three types of selective refinement: hierarchical, progressive, and component. Each of these refinement processes differs in its application, effectiveness, complexity, and amount of memory required.

- Hierarchical selective refinement is used only in the hierarchical mode of operation. It allows for a region of a frame to be refined by the next differential frame of a hierarchical sequence.

- Progressive selective refinement is used only in the progressive mode and adds refinement. It allows a greater bit resolution of zero and non-zero DCT coefficients in a coded region of a frame.

- Component selective refinement may be used in any mode of operation. It allows a region of a frame to contain fewer colors than are defined in the

frame header.

Image tiling

Tiling is used to divide a single image into two or more smaller subimages. Tiling allows easier buffering of the image data in memory, quicker random access of the image data on disk, and the storage of images larger than 64Kx64K samples in size. JPEG supports three types of tiling: simple, pyramidal, and composite.

- Simple tiling divides an image into two or more fixed-size tiles. All simple tiles are coded from left to right and from top to bottom and are contiguous and non-overlapping. All tiles must have the same number of samples and component identifiers and must be encoded using the same processes. Tiles on the bottom and right of the image may be smaller than the designated size of the image dimensions and will therefore not be a multiple of the tile size.
- Pyramidal tiling also divides the image into tiles, but each tile is also tiled using several different levels of resolution. The model of this process is the JPEG Tiled Image Pyramid (JTIP), which is a model of how to create a multi-resolution pyramidal JPEG image.

  A JTIP image stores successive layers of the same image at different resolutions. The first image stored at the top of the pyramid is one-sixteenth of the defined screen size and is called a *vignette*. This image is used for quick displays of image contents, especially for file browsers. The next image occupies one-fourth of the screen and is called an *imagette*. This image is typically used when two or more images must be displayed at the same time on the screen. The next is a low-resolution, full-screen image, followed by successively higher-resolution images and ending with the original image.

  Pyramidal tiling typically uses the process of "internal tiling," where each tile is encoded as part of the same JPEG data stream. Tiles may optionally use the process of "external tiling," where each tile is a separately encoded JPEG data stream. External tiling may allow quicker access of image data, easier application of image encryption, and enhanced compatibility with certain JPEG decoders.
- Composite tiling allows multiple-resolution versions of images to be stored and displayed as a *mosaic*. Composite tiling allows overlapping tiles that may be different sizes and have different scaling factors and compression parameters. Each tile is encoded separately and may be combined with other tiles without resampling.

SPIFF (Still Picture Interchange File Format)

SPIFF is an officially sanctioned JPEG file format that is intended to replace the defacto JFIF (JPEG File Interchange Format) format in use today. SPIFF includes all of the features of JFIF and adds quite a bit more functionality. SPIFF is designed so that properly written JFIF readers will read SPIFF-JPEG files as well.

For more information, see the article about SPIFF.

Other extensions

Other JPEG extensions include the addition of a version marker segment that stores the minimum level of functionality required to decode the JPEG data stream. Multiple version markers may be included to mark areas of the data stream that have differing minimum functionality requirements. The version marker also contains information indicating the processes and extension used to encode the JPEG data stream.

For Further Information About JPEG

The JPEG standard is available in English, French, or Spanish, and as a paper copy or a PostScript or Word for Windows document from the International Standards Organization (ISO) or International Telecommunication Union (ITU). Copies of the standard may be ordered from:

American National Standards Institute, Inc.
Attention: Customer Service
11 West 42nd St.
New York, NY 10036 USA
Voice: 212-642-4900

The standard is published as both an ITU Recommendation and as an ISO/IEC International Standard, and is divided into three parts: Part 1 is the actual specification, Part 2 covers compliance-testing methods, and Part 3 covers extensions to the JPEG specification. Parts 1 and 2 are at International Standard status. See these documents:

> "Digital Compression and Coding of Continuous-Tone Still Images, Requirements and Guidelines," Document number ITU-T T.81 or ISO/IEC 10918-1.

> "Digital Compression and Coding of Continuous-Tone Still Images, Compliance testing," Document number ITU-T T.83 or ISO/IEC 10918-2.

Part 3 is still at Committee Draft status. See this document:

> "Digital Compression and Coding of Continuous-Tone Still Images, Extensions," Document number ITU-T T.84 or ISO/IEC 10918-3.

New information on JPEG and related algorithms is constantly appearing. The majority of the commercial work for JPEG is being carried out at these companies:

Eastman Kodak Corporation
232 State Street
Rochester, NY 14650
Voice: 800-242-2424
WWW: *http://www.kodak.com*
C-Cube Microsystems
1778 McCarthy Boulevard
Milpitas, CA 95035
Voice: 408-944-6300

See the article about the JFIF file format supported by C-Cube and the SPIFF file format defined by Part 3 of the JPEG specification.

The JPEG FAQ (Frequently Asked Questions) is a useful source of general information about JPEG. This FAQ is included on the CD-ROM ; however, because the FAQ is updated frequently, the CD-ROM version should be used only for general information. The FAQ is posted every two weeks to USENET newsgroups *comp.graphics.misc*, *news.answers*, and *comp.answers*. You can get the latest version of this FAQ from the *news.answers* archive at:

ftp://rtfm.mit.edu/pub/usenet/news.answers/jpeg-faq.

You can also get this FAQ by sending email to:

mail-server@rtfm.mit.edu

with the message "send usenet/news.answers/jpeg-faq" in the body.

A consortium of programmers, the Independent JPEG Group (IJG), has produced a public domain version of a JPEG encoder and decoder in C source code form. We have included this code on the CD-ROM. You can obtain the IJG library from various FTP sites, information services, and computer bulletin boards.

The best short technical introduction to the JPEG compression algorithm is:

> Wallace, Gregory K., "The JPEG Still Picture Compression Standard," *Communications of the ACM*, vol. 34, no. 4, April 1991, pp. 30-44.

A more complete explanation of JPEG can be found in the following texts:

> Pennebaker, William B. and Joan L. Mitchell, *JPEG: Still Image Data Compression Standard*, Van Nostrand Reinhold, New York, 1993.

This book contains the complete text of the ISO JPEG standards (DIS 10918-1 and 10918-2). This is by far the most complete exposition of JPEG in existence and is highly recommended.

Nelson, Mark, *The Data Compression Book*, M&T Books, Redwood City, CA. 1991.

This book provides good explanations and example C code for a multitude of compression methods, including JPEG. It is an excellent source if you are comfortable reading C code but don't know much about data compression in general. The book's JPEG sample code is incomplete and not very robust, but the book is a good tutorial.

Here is a short bibliography of additional JPEG reading:

Barda, J.F., "Codage et Compression des Grandes Images," *Proceedings of AFNOR Multimedia and Standardization Conference*, vol. 1, March 1993, pp. 300-315.

Hudson, G., H. Yasuda, and I. Sebestyen, "The International Standardization of Still Picture Compression Technique," *Proceedings of the IEEE Global Telecommunications Conference*, November 1988, pp. 1016-1021.

Leger, A., J. Mitchell, and Y. Yamazaki, "Still Picture Compression Algorithms Evaluated for International Standardization," *Proceedings of the IEEE Global Telecommunications Conference*, November 1988, pp. 1028-1032.

Leger, A., T. Omachi, and T.K. Wallace, "JPEG Still Picture Compression Algorithm," *Optical Engineering*, vol. 30, no. 7, July 1991, pp. 947-954.

Mitchell, J.L., and W.B. Pennebaker, "Evolving JPEG Color Data Compression Standard," in *Standards for Electronic Imaging Systems*, Neir, M. and M.E. Courtot, eds., vol. CR37, SPIE Press, 1991, pp. 68-97.

Netravali, A.N., and B.G. Haskell, *Digital Pictures: Representation and Compression*, Plenum Press, New York, 1988.

Rabbani, M., and Jones, P., *Digital Image Compression Techniques, Tutorial Texts in Optical Engineering*, vol. TT7, SPIE Press, 1991.

This page is taken from the Encyclopedia of Graphics File Formats and is licensed by O'Reilly under the Creative Common/Attribution license.

Feedback Printer friendly

Terms of Service | Privacy Policy | Contact Info

Second Declaration of George T. Ligler

Exhibit 18

Raw vs. JPEG: Style or Substance?

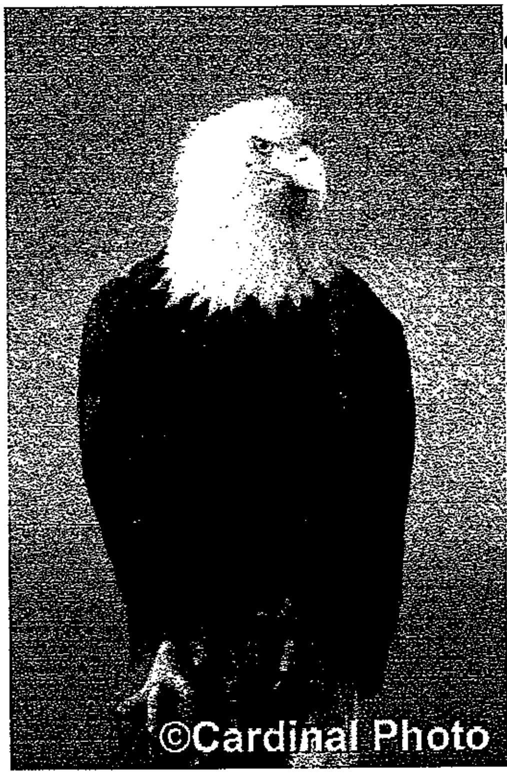
The very first thing to know about Raw, TIFF, and JPEG is that if you have a pro quality D-SLR (this includes at least the Canon D60, 10D, 1D, 1Ds and the Nikon D100, D1, D1X and D1H) you can take high-quality publishable photographs in any of the 3 file formats (although you'll want to be using the highest quality settings and highest resolution available for whichever format you choose). All D-SLRs have hundreds of dollars worth of electronics--that you paid for--which are dedicated to taking the raw sensor data and turning it into a useable finished image in a fraction of a second. This is probably the most remarkable miracle of modern digital camera technology. If your camera vendor claims you need to shoot Raw format to get the best quality from their camera, ask them if that means you've wasted $500 or more paying for all those extra circuit boards to do in camera image processing:-) Or if they don't have confidence in their own camera to produce a finished image. Frankly, its just as likely that they have a vested interest in locking you in to their proprietary file format and image processing software system. We'll come back to the quality issue, but first we'll look at the broader issue of shooting Raw vs. JPEG.

If we clear away all the computer jargon the Raw vs. JPEG discussion is really the same as one that has been going on in photography for decades. Basically it has to do with when your photograph is finished. Any one who shot chromes, particularly for editorial submission or projection, was used to the image being complete at the moment the shutter was pressed. As long as the light was within the range of the film and the metering was set the way you wanted, the photo was finished. If you needed to print a slide you could send it off to a lab and the lab tech worked to match the chrome. By contrast, many photographers who shot with negative film were used to an extended creative process that might involve hours in the darkroom working to finish the image that they started in the camera. The greater latitude and resulting lower gamma (contrast) of negative film meant that an extra creative step was needed to print the final image. And since negatives can't be viewed directly there wasn't any single "right" way to visualize the image contained on a negative. Obviously the most famous example of the importance of printing were the many famous Ansel Adams images that were of only passing interest until reprinted many years later in a way that caught the viewer's attention.

Today we are witnessing a replay of this same phenomenon with digital. Depending on your job requirements or your style, you may work to create the image in the camera. If so, then the camera's extensive image processing circuitry is your best friend. It'll will turn the light you capture into a finished and viewable image practically before you let up on the shutter. Just as with chromes, your image is complete and ready to use. Of course this assumes you've got the lighting, exposure and white balance just the way you want them before you press the shutter. On the other hand, if you think of the scene in front of you as the raw material and yearn for the time you spent performing your special magic in the darkroom then Raw files give you back that same flexibility. You can remap the 12-bits of information captured by the camera into the final 8-bits needed for output. You can change the white balance to your heart's content and you can even alter the exposure by several stops without ruining the image. You can do some of that with JPEGs, but not as well or as extensively.

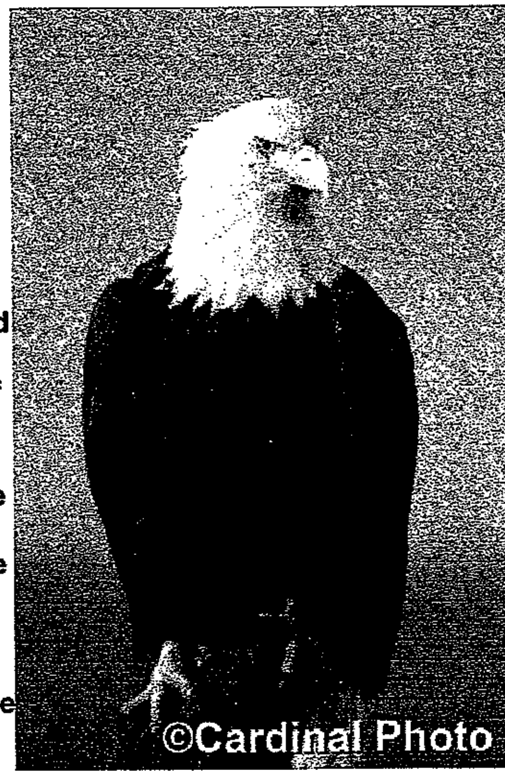
If everyone was willing to just leave the choice at that life would be fairly simple. Raw files allow you to create a digital darkroom and fulfill your creative yearnings at the computer while JPEGs complete the image at the time of capture and recreate the freeing feeling of chromes when your image is finished at the time you click the shutter. But in the interests of selling seminars or software, or just promoting their workflow there is a steady siren's song that claims you need to shoot Raw files to get the most from your camera. Frankly, that's bunk. Sure, if you live in the world of numbers, there is good, solid math that shows you can tweak more decimal points out of a Raw file after the fact. That 2GHz Pentium and enough of your time can do a technically superior job when compared to the VLSI in the camera and the 100 milliseconds it has to do its work. But that comparison completely ignores your skill as a photographer, your needs as a working pro, and the value of your time. If you are familiar with your subject and the lighting conditions, you can often capture the image so that it can be directly processed into a JPEG without losing anything you wanted it to communicate. And if you are paid to produce quality images, the increased size of Raw images and the increased processing time required will decrease your productivity. That's fine if you need Raw images to create your final result, but if you don't then Raw files are just slowing you down. Finally, your time. If you enjoy working with your images, just like you enjoyed working in the darkroom, then great. But if not, then let the camera you paid all that money for do the work rather than making you do it! If you can live with the larger file size, Canon's Raw+JPEG option is a very creative approach to providing the best of both worlds. Kodak's ERI format is as well. Nikon's attempt to emulate this capability by adding JPEG creation for Raw files in Capture is nice in theory but defeats the entire point of having the image finished in the camera by requiring the purchase and use of their software to create the JPEG. (Note: Unfortunately with the Canon 10D, it appears that special software support is now

needed to extract the JPEG image from the Raw+JPEG file. This removes much of the advantage of having the JPEG there in the first place.)



This image of a Bald Eagle works as is, shot with a White Balance of 6510K.

If I'd wanted the Eagle to look as if it was in front of a perfect blue sky I could have shot with a lower color temperature setting.



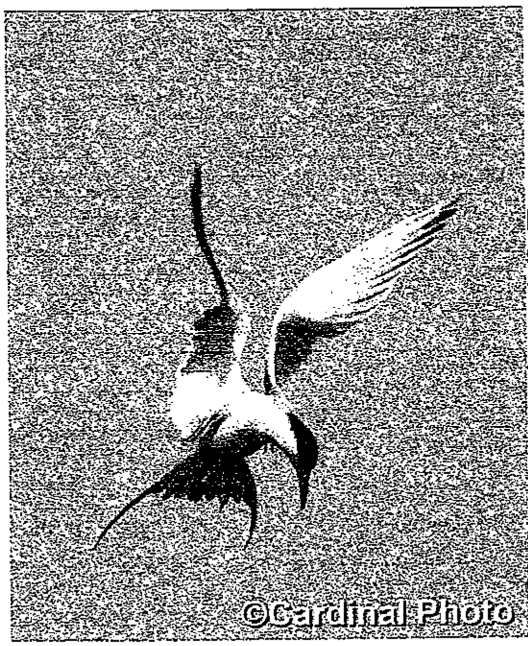
But since the imag is a NEF file I can easily change the Whit Balance after the fact in Adobe Camera Raw to create the classi blue sky look if I change my mind later.

The imag isn't any better as a result, but I do have moi freedom to make choices "in the darkroon instead o just behind th camera.

Okay, so by now you're saying, "But I heard that you need to shoot Raw to get the best images". Let's talk about that in more detail. The first thing you've probably been told is that Raw files are 12-bits and JPEG files are 8-bits so they're "better". However that is apples vs. oranges. The 12-bit CCD is a linear encoding of light. Each of the 4096 levels represents an equal number of photons. The human visual system's response to light isn't linear though. Humans are sensitive to percentage changes in light, not absolute changes. So at the high end of the 4096 values there is "too much" information (the brightest stop of light is represented by 2048 separate values, from 2048 to 4095) while at the low end of the 4096 values there is "too little" information (the darkest stop of light is represented by only 2 values, 1 and 2!). Nikon itself uses this fact to create visually lossless compressed NEFs. The compressed NEF format uses just over 500 values (a little over 9-bits) to encode all the data in a NEF file by spreading the values out in a way that more accurately reflects the sensitivity of the human eye. The camera uses a tone curve (essentially

a modified gamma curve) to push that a little further, into an 8-bit (256-value) encoding. Sure there is some small loss of mathematical detail, but if the tone curve is the right one for the image and you don't need to move the exposure after the fact you'll be hard pressed to see the difference. And since essentially every output device in common use is 8-bits or less per color, your software is going to need to do convert the image to 8-bits per color at some point in any case. The only advantage of doing it on the computer is that you can change your mind or re-purpose the image after the fact by doing it several different ways.

A very similar argument is also heard about JPEG compression. There is a gnawing fear that somehow because JPEG is a "lossy" compression algorithm you'll be throwing away that vital pixel. The truth is that modern JPEG compression–when used with the High quality settings found in modern D-SLRs or Photoshop–is essentially visually lossless if used only once or twice on an image. There is a detailed comparison of different levels of compression and how they specifically affect pixel values in the D100 and D1 Generation Update eBook if you're interested in the gory details. There is no question that by forcing the choice of exposure, tone curve, and white balance at the time of capture JPEGs limit your options to change your mind after the fact–just like slide film did. But that doesn't mean that slides aren't as good as negatives or that JPEGs aren't as good as Raw files. It means you need to decide on your goals for your photography and about which format fits your shooting needs and style.



By shooting JPEG I can capture a higher percentage of split second action shots like this Forster's Tern hovering while it looks for fish.

And by setting the exposure and white balance right when I capture the image I don't have to do any work in the darkroom to get the final image I want.

For me personally, the decision is normally easy. I spend too much time in the office or on the computer as it is. I'd rather invest my effort in learning how to capture the image the way I want it when I press the shutter than in perfecting my digital darkroom technique. And photographing wildlife brings in a whole other dimension. The constant activity of the subject means that JPEG offers the substantial advantage of letting me capture more of the action more of the time. Any image I capture is of course better quality than one I don't! But, in case you think we don't spend enough time with Raw files to truly appreciate their beauty, in the process of creating DigitalPro my co-developers and I have shot and built decoders and viewers for over a dozen different variants of Nikon and Canon Raw file formats, so we a solid understanding of what those file formats offer and how D-SLRs and computer software alike render them into completed images.

But, as my daughter says, "Make your own opinion". There are as many opinions out there as there are photographers, but only yours matters. What's important is that you have confidence in your ability, your equipment and your workflow to help you produce the images you need when you need them. Have fun and good shooting!–David

We'd love to hear your opinion, as well as what format or formats you shoot and how they help you creatively and in your business. Please join us on the DigitalPro forums and let us know!

AX205636

Second Declaration of George T. Ligler

Exhibit 19

Kodak Share Moments. Share Life.

Digital Learning Center

Return to DLC Main

How To Chapter I, Creating Digital Images: Lesson IV

Keep File Size Small

When you capture images, it is best to work with the highest resolution and least compression possible because the resulting images give you the most "information" to work with as you enhance them. This initial image file is most often over one megabyte in size.

Once the image has been enhanced, you will want to save it in its highest version - as a reference master. Then you will want to resize it, sharpen it and save it in a format that displays well on a web page and is the smallest size possible. Some design experts recommend that no graphic file (whether digital image or graphical art work) be larger than 50k in size. Remember, people viewing web pages have plenty of other places to surf if your pages are loading too slowly!! Since design standards for this new media format are still evolving (as is the technology), it is too soon to set hard and fast rules. The following guidelines will be helpful as you set up your own files.

1. Save photographic images in JPEG format. JPEG (pronounced "jay-peg") is an acronym that refers to the name of the committee who wrote the standard for this particular image compression format.
2. When you save in JPEG format, compare the images and you may find that a medium resolution option will look as good on screen as a higher one. (The lower the resolution, the smaller the file.)
3. Always save a master image without compression so you have a high quality image to use as a starting point.
4. Experiment with the amount of JPEG compression. More compression will reduce display quality. Less compression will retain display quality. Find the maximum amount of compression that still delivers good results.
5. If your imaging software does not give you options for saving JPEG files at various resolution options, your files may be quite large. In that case, the only way you can reduce file size is by cropping the image. If your web site relies extensively on digital images, you may want to consider obtaining other imaging software that

AX202421

Kodak Share Moments. Share Life.
Digital Learning Center

Return to DLC Main

How To *Chapter I, Creating Digital Images: Lesson IV*
Keep File Size Small

When you capture images, it is best to work with the highest resolution and least compression possible because the resulting images give you the most "information" to work with as you enhance them. This initial image file is most often over one megabyte in size.

Once the image has been enhanced, you will want to save it in its highest version - as a reference master. Then you will want to resize it, sharpen it and save it in a format that displays well on a web page and is the smallest size possible. Some design experts recommend that no graphic file (whether digital image or graphical art work) be larger than 50k in size. Remember, people viewing web pages have plenty of other places to surf if your pages are loading too slowly!! Since design standards for this new media format are still evolving (as is the technology), it is too soon to set hard and fast rules. The following guidelines will be helpful as you set up your own files.

1. Save photographic images in JPEG format. JPEG (pronounced "jay-peg") is an acronym that refers to the name of the committee who wrote the standard for this particular image compression format.
2. When you save in JPEG format, compare the images and you may find that a medium resolution option will look as good on screen as a higher one. (The lower the resolution, the smaller the file.)
3. Always save a master image without compression so you have a high quality image to use as a starting point.
4. Experiment with the amount of JPEG compression. More compression will reduce display quality. Less compression will retain display quality. Find the maximum amount of compression that still delivers good results.
5. If your imaging software does not give you options for saving JPEG files at various resolution options, your files may be quite large. In that case, the only way you can reduce file size is by cropping the image. If your web site relies extensively on digital images, you may want to consider obtaining other imaging software that

AX202428

Kodak Share Moments. Share Life.

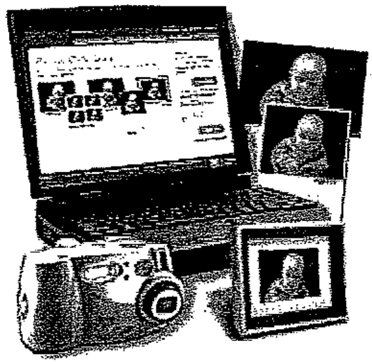
Digital Learning Center

Return to DLC Main

How To *Chapter VIII*

Online Printing

Get amazing color prints from your digital pictures.



One touch.
1. Upload your digital pictures.

Get Prints Online!
2. Share, store & enhance your photos online.

3. Get amazing quality prints & photo gifts.

Now the same Kodak quality prints you're used to getting from your traditional film camera are available online for your digital camera prints—and for about the same price. With EasyShare, uploading your pictures to online photo services is really easy, and the prints you receive in the mail have amazing quality. You can also:

- Store and organize your photos in secure online albums
- Easily share with friends and family
- Get creative with easy editing tools and photo borders
- Shop for frames, cards, and albums

To find where you can have your pictures printed online, visit www.kodak.com.

Begin Lesson...

Return to Book Start...
Return to DLC Main...

DLC Plus

AX202488

Kodak Share Moments. Share Life.

Digital Learning Center

Return to DLC Main

How To

Chapter VIII, Online Printing: Lesson I

Getting Ready to Take Pictures

Page 2 of 3

What Is Image Compression?

Photographic pictures generate large amounts of data. For example, one 35 mm scanned negative can be an 18 megabyte file. If that were text, it would fill over 6000 pages.

In order to handle digital pictures more efficiently, they are compressed. Some methods of compression are LOSSLESS. That is, the compressed file will result in a picture that , when uncompressed, is identical to the original. These methods generally don't reduce file size by more than a 2:1 ratio.

To achieve higher compression levels the methods used are referred to as LOSSY. That is, they sacrifice some of the original image data in order to achieve the significant reduction in file size. Moderate levels of compression, those less than a ratio of 10 to 1, are often used with good results. Compression levels greater than 10 to 1 generally result in a reduction of print quality.

Low Compression Example

High Compression Example

AX202491

Return to DLC Main

How To

Chapter VIII, Online Printing: Lesson I

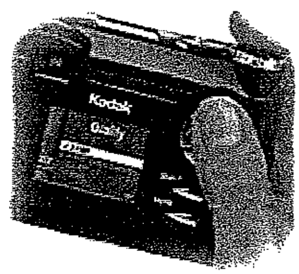
Getting Ready to Take Pictures

Page 3 of 3

What can I do to prepare my digital camera pictures for the best color and quality results?

Color quality begins with the picture capture. To optimize the capture, use the following tips:

- Review your camera Owner's Manual to understand the best setting for your camera in the lighting conditions you are using (daylight, fluorescent, incandescent).
- Set the Quality setting on your camera to capture a sufficient number of pixels for your prints.
- Use the largest native resolution your camera has. Avoid using "enhanced" or "interpolated" resolutions.
- If you are using a picture editing application, avoid "sharpening" the picture.
- When saving the picture, if you have the choice, select RGB as the default color space for your pictures. Check the User's Manual for your application for help in selecting the color space.
- Save your picture in 24-bit color.
- If you are using a JPEG compression or a quality setting, select the Best setting to ensure the least amount of compression and larger file size.



Previous **Next Lesson**

Return to Chapter Start...
Return to Book Start...
Return to DLC Main...

DLC Plus

Home | Privacy | Website Terms of Use

AX202493

Return to DLC Main

Technology

Chapter I

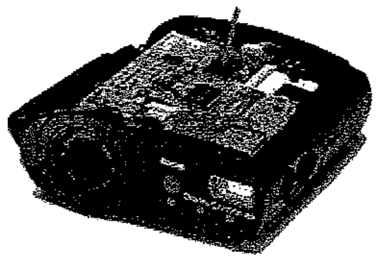
What's Inside

Lessons in this chapter

Chapters in this book

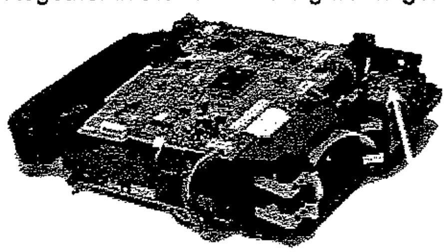
4 Process the image

The pixels start moving off the CCD row by row in a serial fashion (one after the other). They pass through many of the components in the camera for white balance, color, and aliasing correction. The pixels continue on until they get to the 4MB Frame board (indicated by the red arrow).



5 Put the image back together

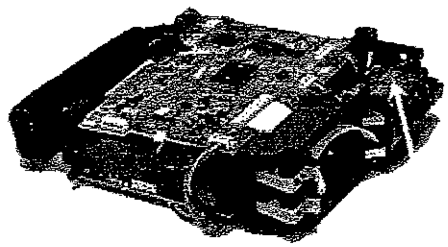
The 4MB Frame board (indicated by the yellow arrow) puts all the pixels back together in the form of a digital image.

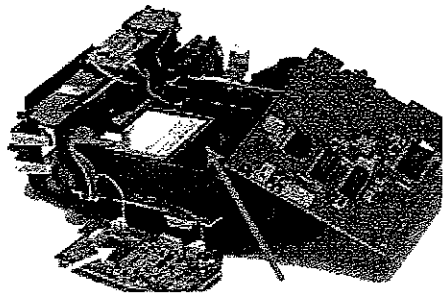


6 Compress the image

As the image continues on its journey it leaves the 4MB Frame board (yellow arrow) and is compressed. The compression applied is determined by the user before they capture the image. There are two compression options: Snapshot (aggressive compression) or High (minimum compression).

AX202503





Continue Lesson...
Return to Chapter Start...
Return to Book Start...
Return to DLC Main...

DLC Plus

Home | Privacy | Website Terms of Use

AX202504

Kodak TAKE PICTURES. FURTHER.™
Digital Learning Center

Return to DLC Main

Technology

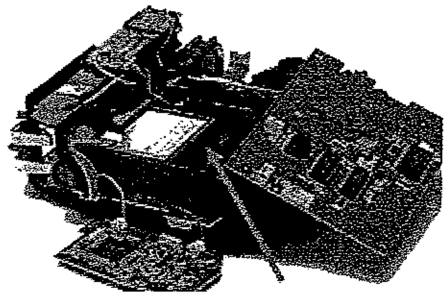
Chapter I
What's Inside

Lessons in this chapter

Chapters in this book

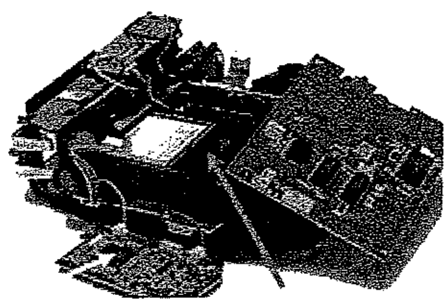
7 Long-term storage

The "project manager" I(MCU chip) sends out a message to move the compressed image to the 1MB flash RAM (magenta arrow) which is used for long-term storage.



8 Adjust LCD display

When the image safely reaches the 1MB flash RAM (magenta arrow) the project manager sends a message to the 4-bit MPU (yellow arrow) to reduce the "Pictures Remaining" field on the LCD by one.



Next Lesson...
Return to Chapter Start...
Return to Book Start...
Return to DLC Main...

DLC Plus

Home | Privacy | Website Terms of Use

AX202505

Kodak TAKE PICTURES. FURTHER.™

Digital Learning Center

Return to DLC Main

Technology

Chapter IV, Image File Format Overview: Lesson IV

Other File Formats

The GIF File

GIF -- or Graphics Interchange Format -- files define a protocol intended for the on-line transmission and interchange of raster graphic data in a way that is independent of the hardware used in their creation or display.

The GIF format was developed in 1987 by CompuServe -- one of the world's most successful bulletin board services -- for compressing eight-bit images that could be telecommunicated through their service and exchanged among users.

The GIF file is defined in terms of blocks and sub-blocks which contain relevant parameters and data used in the reproduction of a graphic. A GIF data stream is a sequence of protocol blocks and sub-blocks representing a collection of graphics.

The JPEG File

JPEG is a standardized image compression mechanism. The name derives from the Joint Photographic Experts Group, the original name of the committee that wrote the standard. In reality, JPEG is not a file format, but rather a method of data encoding used to reduce the size of a data file. It is most commonly used within file formats such as JFIF and TIFF.

JPEG File Interchange Format (JFIF) is a minimal file format which enables JPEG bitstreams to be exchanged between a wide variety of platforms and applications. This minimal format does not include any of the advanced features found in the TIFF JPEG specification or any application specific file format.

JPEG is designed for compressing either full-color or grayscale images of natural, real-world scenes. It works well on photographs, naturalistic artwork, and similar material, but not so well on lettering or simple line art. It is also commonly used for on-line display/transmission; such as on web sites. A 24-bit image saved in JPEG format can be reduced to about one-twentieth of its original size.

The Photoshop File

A Photoshop file is the native file format for Adobe Photoshop. A file saved in this manner can only be opened and edited in Photoshop. However, the user has the option to save the file in a variety of other formats that are readable in both the Macintosh and PC environment.

The major advantage of the Photoshop format becomes apparent when working on documents with layers. For example, a background can be created on one layer, then graphics can be added on a second layer, a drop-shadow on a third layer and text on yet another layer. Each layer is independent of the others and can be edited separately without affecting the contents of the other layers. Photoshop

AX202563

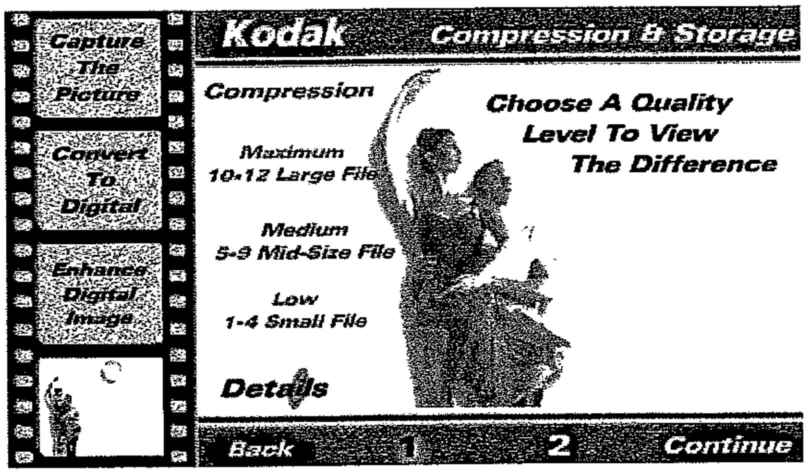
Return to DLC Main

Technology

Chapter VI

Path of Digital Photography

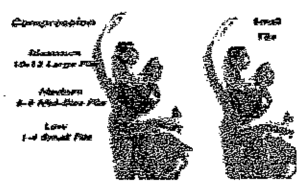
Storing and Saving Your Digital Pictures



This section leads you through the last steps taken before you can start using and sharing your pictures.

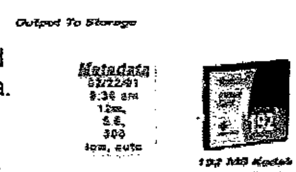
1 Compressing Your Picture Files

The size of a 3.1 megapixel, bit mapped image is over 9 megabytes. To stretch the capacity of your Picture Card, The JPEG compression standard is used. This decreases the file size to an average of 850KB. During image compression, the image processor selectively filters the image to produce a smaller image file with minimal impact on picture quality.



2 Creating The Finished Image File

The JPEG compressed image file is combined with a thumbnail image for quick previewing and detailed capture information known as Metadata. Then, the image is saved to the removable Picture Card. These pictures subsequently can be transferred to a computer, television or other device for viewing, editing, sharing and archiving.



3 Reviewing The Process

If you'd like to view the whole digital capture process again and see how each of these four steps fits together along the Path to Digital, click here!

Competency Exam

AX202584

Kodak Share Moments. Share Life.

Digital Learning Center

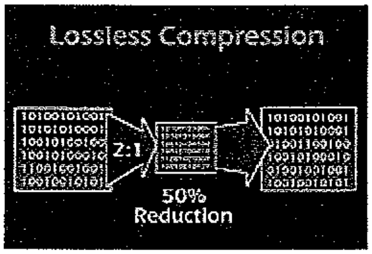
[Return to DLC Main](#)

Courses *Chapter I, Digital Imaging Fundamentals: Lesson III*

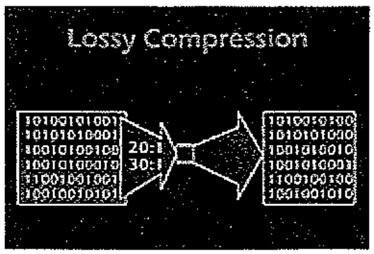
Processing

Page 9 of 14

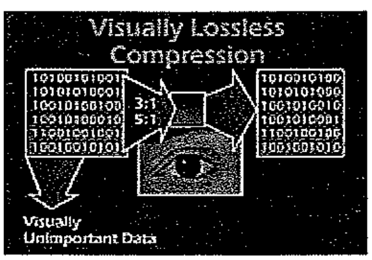
There are two basic types of data compression: lossless compression and lossy compression.



Lossless compression achieves only about a 2:1 compression ratio, but the reconstructed image is mathematically and visually identical to the original.



Lossy compression provides much higher compression rates, but the reconstructed image shows some loss of data compared to the original image. This loss can be visible to the eye or visually lossless.



Visually lossless compression is based on knowledge about color images and human perception.

Visually lossless compression algorithms sort image data into "important data" and "unimportant data," then discard the unimportant.

Previous **Continue**

Return to Chapter Start...
Return to Book Start...
Return to DLC Main...
DLC Plus

Kodak Share Moments. Share Life.

Digital Learning Center

[Return to DLC Main](#)

Courses

Chapter III, Desktop Imaging Systems and Issues: Lesson IV

Working With Images

page 5 of 10

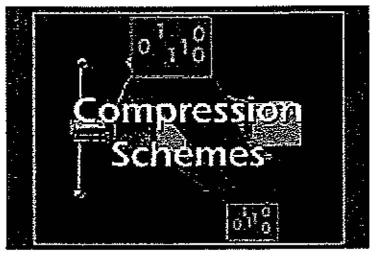
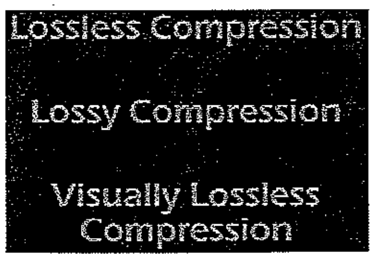
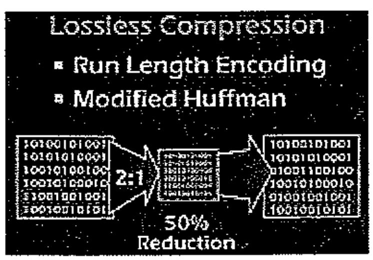


Image compression uses mathematical techniques to allow an image to be stored in less memory by removing redundancies in image data.



There are three basic types of image compression: lossless, lossy, and visually lossless.



Lossless compression achieves only about a 2 to 1 compression ratio, but the reconstructed image is mathematically and visually identical to the original.

Previous **Continue**

Return to Chapter Start...
Return to Book Start...
Return to DLC Main...

DLC Plus

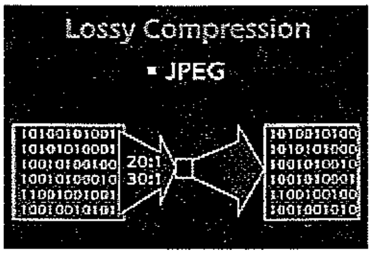
AX202792

Return to DLC Main

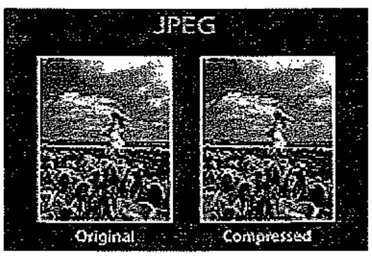
Courses *Chapter III, Desktop Imaging Systems and Issues: Lesson IV*

Working With Images

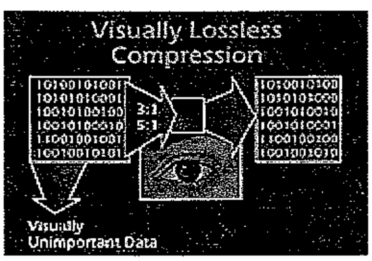
page 6 of 10



Lossy compression provides much higher compression rates, but the reconstructed image shows some loss of data compared to the original image.



JPEG, a common lossy compression scheme, offers a range of compression ratios. Depending on image content, an image file can be compressed to about one tenth the original size without noticeable degradation.



Visually lossless compression, based on information about color images and human perception, sorts image data into "important data" and "unimportant data," then discards the unimportant data.

Previous **Continue**

Return to Chapter Start...
Return to Book Start...
Return to DLC Main...
DLC Plus

Home | Privacy | Website Terms of Use

AX202793

Second Declaration of George T. Ligler

Exhibit 20

CONFIDENTIAL EXHIBIT